Automated Test Equipment Simulation In CoppeliaSim Using C# Over WebSocket

Sudip Chakraborty [1] & P. S. Aithal [2]
[1]D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India,
OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in
[2]ViceChancellor, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

Area/Section: Computer Science.
Type of the Paper: Experimental Research.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed in: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.8117650
Google Scholar Citation: IJMTS

How to Cite this Paper:
Chakraborty, S., & Aithal, P. S. (2023). Automated Test Equipment Simulation In CoppeliaSim Using C# Over WebSocket. *International Journal of Management, Technology, and Social Sciences (IJMTS), 8*(2), 284-291. DOI: https://doi.org/10.5281/zenodo.8117650

International Journal of Management, Technology, and Social Sciences (IJMTS)
A Refereed International Journal of Srinivas University, India.

CrossRef DOI: https://doi.org/10.47992/IJMTS.2581.6012.0283

Received on: 05/06/2023
Published on: 30/06/2023

Automated Test Equipment Simulation In CoppeliaSim Using C# Over WebSocket

Sudip Chakraborty [1] & P. S. Aithal [2]

[1]D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India,

OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in

[2]ViceChancellor, Srinivas University, Mangalore, India,

OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

ABSTRACT

Purpose: Nowadays, we are surrounded by electronic gadgets. Most gadgets have more or less printed circuit boards (PCB). It needs to test before it is launched into the market. Otherwise, it can fail to sustain itself in the market. If we test manually in all aspects, a motherboard will take so many times that the production will go very slowly. So automated tests are becoming popular. There is various vendor that produces automated test equipment. ATE or automated test equipment is too costly. Purchasing the ATE system for training or learning the process flow is sometimes not feasible. This research work demonstrates a simulation framework to view the ATE working procedure. We use CoppeliaSim for animation and C# to control it over Websocket. All source codes are available. The researcher can customize and extend the functionality to their need.

Design/Methodology/Approach: We developed a framework to demonstrate the ATE activity. It consists of CoppeliaSim Simulation software. We use C# language to control the simulator. At first, we created a model of ATE inside the CoppeliaSim. Add LUA code to receive commands from the remote host. When CoppeliaSim starts, it starts a WebSocket server. It listens to port number 8088. Then we created an application using C#. We designed a graphical user interface (GUI). We added some text boxes, buttons, and a WebSocket client. When connect button is pressed, the apps try to connect with CoppeliaSim. Then send the command by pressing the button.

Findings/Result: To understand the ATE working principle, we developed a simulation in CoppeliaSim, which controls using a C# websocket client. The simulation works like a natural system. Observing the process flow, the researcher gets an understanding better. So using this example and customizing it according to the need is the best way to understand it.

Originality/Value: Several procedures are available to understand the ATE system. Here we developed a simplistic approach to it. The CoppeliaSim is suitable for visualizing process automation. The C# language is better for controlling the simulation. The communication between two software is websocket which is easy to integrate, and send/received command string is also comparatively easy. The researcher trying to understand or work with the ATE system can get some reference information for their research work. So the complete framework process is the best value to the researcher.

Paper Type: Experimental-based Research.

Keywords: ATE Simulation, Automated Test Equipment in CoppeliaSim, WebSocket in CoppeliaSim, C# communication over Websocket

1. INTRODUCTION :

Surviving without automated test equipment (ATE) is challenging, especially in the electronics industry. Every electronics product, especially high-end and complex products, uses a motherboard or different accessories board, which must be tested before the product goes to the market. Testing around all aspects takes time. That is why automated testing is gradually becoming popular.

The system tests the board by moving the board automatically. At first, the board is fed to the system. The system takes the board and sends it to the docking station. The docking station then moves the testing board. The handler is connected to several pogo pins. The pogo pins are nothing but some connection point. It is a particular type of pin whose recycle times are very high. When the board is placed in an indicative place, The handler comes down on the board. Several connections are connected. After a successful connection system starts checking the board. The ATE system creates an environment so that motherboard assumes it is installed in a natural system. Whatever signal needs to operate the motherboard, the system provides. The system configuration is changed if the motherboard is changed because different motherboards need different signals to work.

Here we designed a model which mimics the ATE system. It has two sides. One side is to display the ATE-simulated hardware, and the other side is a controller which controls the simulation. We developed the controller in C#. We use websocket communication between the simulator and C#. When we press the button, the command string sends to the simulator and moves the system based on the command we are sending.

2. RELATED WORKS :

In their paper, Miller, L. W. et al. introduce DYNA-SIM, a nonstationary queuing simulation, and apply it to the automated test equipment problem. It presents a mathematical model and simulation approach to analyze the performance and efficiency of automatic test equipment systems [1]. Montiel, C. M focuses on a folded planar Marchand balun design for radio frequency automated test equipment applications. It addresses the challenge of improving isolation in balun designs for accurate R.F. measurements [2]. Barnes, H. et. Al., in their paper, explores the development of a pogo pin assembly via design for multi-gigabit interfaces on automated test equipment [3]. Boot, R., in their paper, presents a computerized test approach for electronic control units (ECUs) in a hardware-in-the-loop simulation environment [4]. Jiang, C. et al.'s paper investigates real-time test script development for mechanical test equipment. It explores the challenges and considerations in designing test scripts that enable efficient and reliable test execution [5]. Moreira, J., in their paper, discusses the passive equalization of test fixtures for high-speed digital measurements using automated test equipment [6]. Liguori, F. introduces computer aids for digital test design in automatic test equipment. It discusses using computer-based tools to enhance the efficiency and effectiveness of test design processes [7]. Wang, Q., et. Al., in their paper, introduces a cloud-based simulated automated testing platform for virtual coupling systems [8]. Gwóźdź, M. focuses on developing a current reference generator for high-power mechanical test equipment. It addresses the challenge of generating accurate and stable reference currents for testing high-power devices [9]. In their paper, Zaidi, Y. et al. discuss rapid automated testing, verification, and validation for CubeSats, focusing on the challenges and solutions for testing small satellite systems [10].

3. OBJECTIVES :

The objectives of the research Designing and implement a framework that accurately emulates the behaviour of actual test equipment within the CoppeliaSim simulation environment. It enables engineers to interact with the simulation, modify parameters, and retrieve data. Ensure that engineers can monitor and control the simulation in real time, facilitating efficient test execution and immediate feedback. It demonstrates the framework's cost-effectiveness and time efficiency by reducing reliance on physical testing, minimizing setup and maintenance costs, and accelerating the time-to-market process for electronic systems. Design the framework to be flexible and scalable, allowing engineers to simulate a wide range of test equipment configurations and scenarios. Enable easy integration of additional test equipment components to accommodate diverse testing requirements.

4. APPROACH AND METHODOLOGY :

The complete project block diagram is depicted in Figure 1.

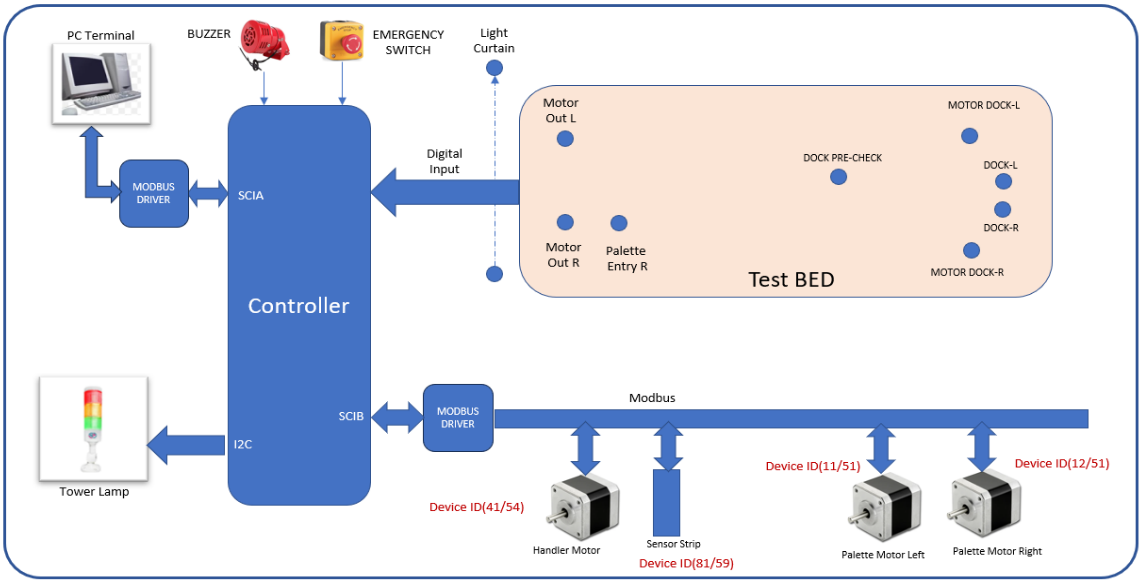

Fig. 1: Project Block Diagram [Source: Author's]

Now we will discuss the block one by one.

1. **P.C. Terminal**: it is a user interface. We send the command to the controller to execute the command. A couple of commands are as below:-
   - **DOCK**: This command is used to send dock commands. The dock command means the DUT(device under test) will go under the Pogo pin, and the top handler will come down to touch the nail. The system sends a series of various signals to test the device.
   - **UNDOCK**: This is the opposite action of the dock. After the test, the user sends the command to undock so that the DUT returns to the operator and places another pallet to test.
   - **HANDLER DOWN**: It brings the handler down
   - **HANDLER UP**: it brings the handler up.
   - **FORCE OUT**: if the pallet is stuck, it is used to force it out.
   - **GO HANDLER:** if any desired position needs to go, we use this command
   - **PUSH DUT:** to simulate, it is used to move the DUT.
   - **DEMO:** The button is used to run the demo for understanding.
2. **Controller**: It is the heart of the system. It controls the complete system. It contains the communication, reads the input, sends the necessary signal to the motor, and responds to the host.
3. **Tower Lamp**: To convey a message visually, One tower lamp is connected. It has potential free contact. We trigger through relay for operation.
4. **Test Bed**: The test bed is the surface where all pallet motor movement mechanism is established.
5. **Pallet Motor**: The motor moves the DUT (Device Under Test) according to the Y axis. It carries the pallet under the handler so the test pin can contact DUT. It is built with two motors. Every motor is Modbus enabled. When the Modbus master sends the signal, it receives and starts/stops the motor.
6. **Handler Motor**: This stepper motor is responsible for movement up/down. This motor is Modbus enabled. It has a unicast and multicast Modbus ID. Unicast is its message and response back to the master. However, it keeps silent for multicast and acts according to the message content.
7. **Sensor Strip**: The sensor strip detects the position of the z-axis movement. This means handler up/down movement is measured using a sensor optoisolator. The sensor is an optically reflected type.
8. **Buzzer**: The buzzer or audio signal alerts the operator or informs some information. After powering up, the controller will start all connected modules. It should initialize to work. When all modules are initialized and the system is ready to use, it sends three beeps, which means the system is prepared to use. When some error occurs, it will sound error beep. On the emergency switch, a long beep sound indicated something seriously happened.

9. **Emergency Switch**: the emergency switch is essential for every instrument. This switch is connected with an interrupt of the controller. It is situated in the most convenient place, making any circumstance easily accessible.

5. EXPERIMENT :

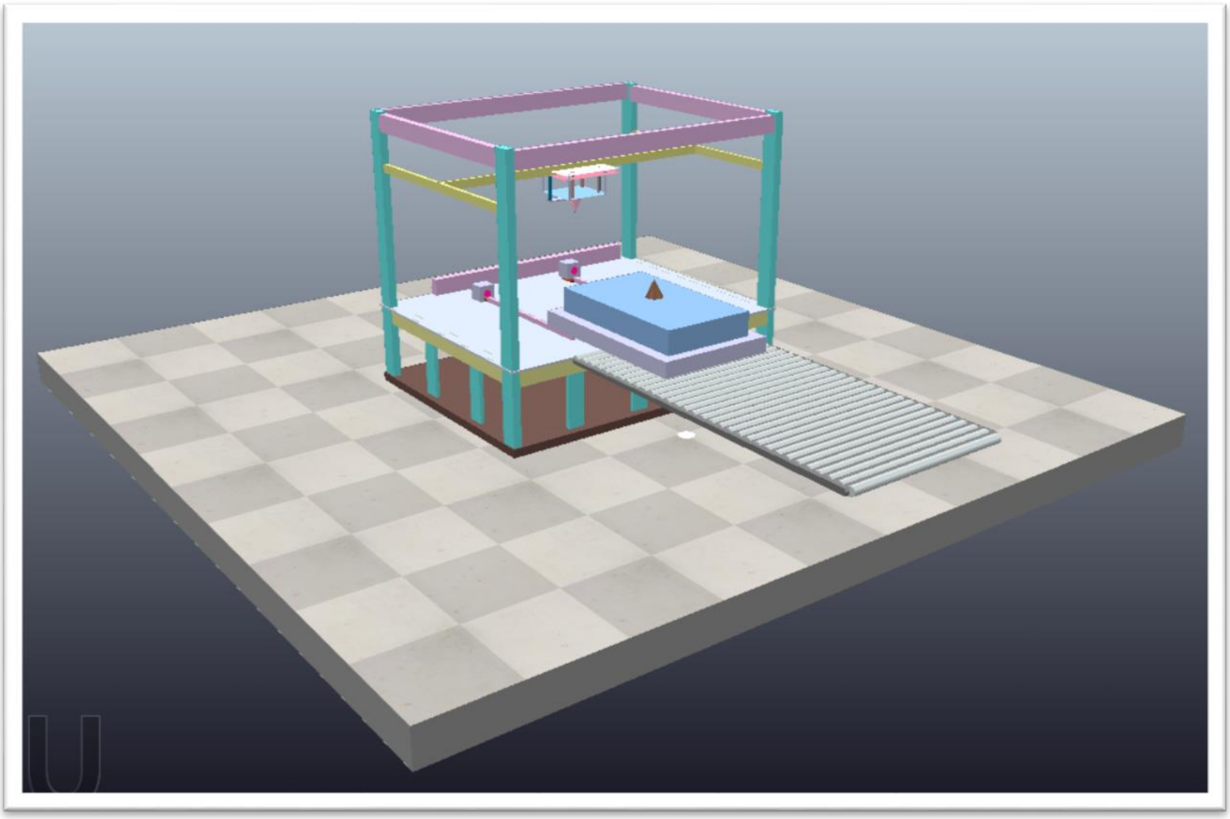

Fig. 2: Simulation Environment [Source: Author's]

Now we can see what is happening when we run the simulation. We need to follow the following steps:

1) Download and install the CoppeliaSim from https://www.coppeliarobotics.com/downloads.
2) Download and install the Microsoft visual studio community edition from https://visualstudio.microsoft.com/downloads/
3) Download the project code from https://github.com/sudipchakraborty/Automated-Test-Equipment-Simulation-In-CoppeliaSim-Using-C-sharp-Over-WebSocket.git.
4) Under the CoppeliaSim folder, double-click on "**TB_Fixture.ttt**". It will open a scene like Figure 2.
5) If we want only to control the simulation, double-click on the GUI_Commander/ bin/Debug/ GUI_Commander.exe file. It will open a GUI depicted in Figure 3. Then press the connected button, and under the "DEMO" group box, click "**START**." We will observe that the simulation is running inside the CoppeliaSim IDE.
6) If we are interested in GUI modification, open "**GUI_Commander. sln**".

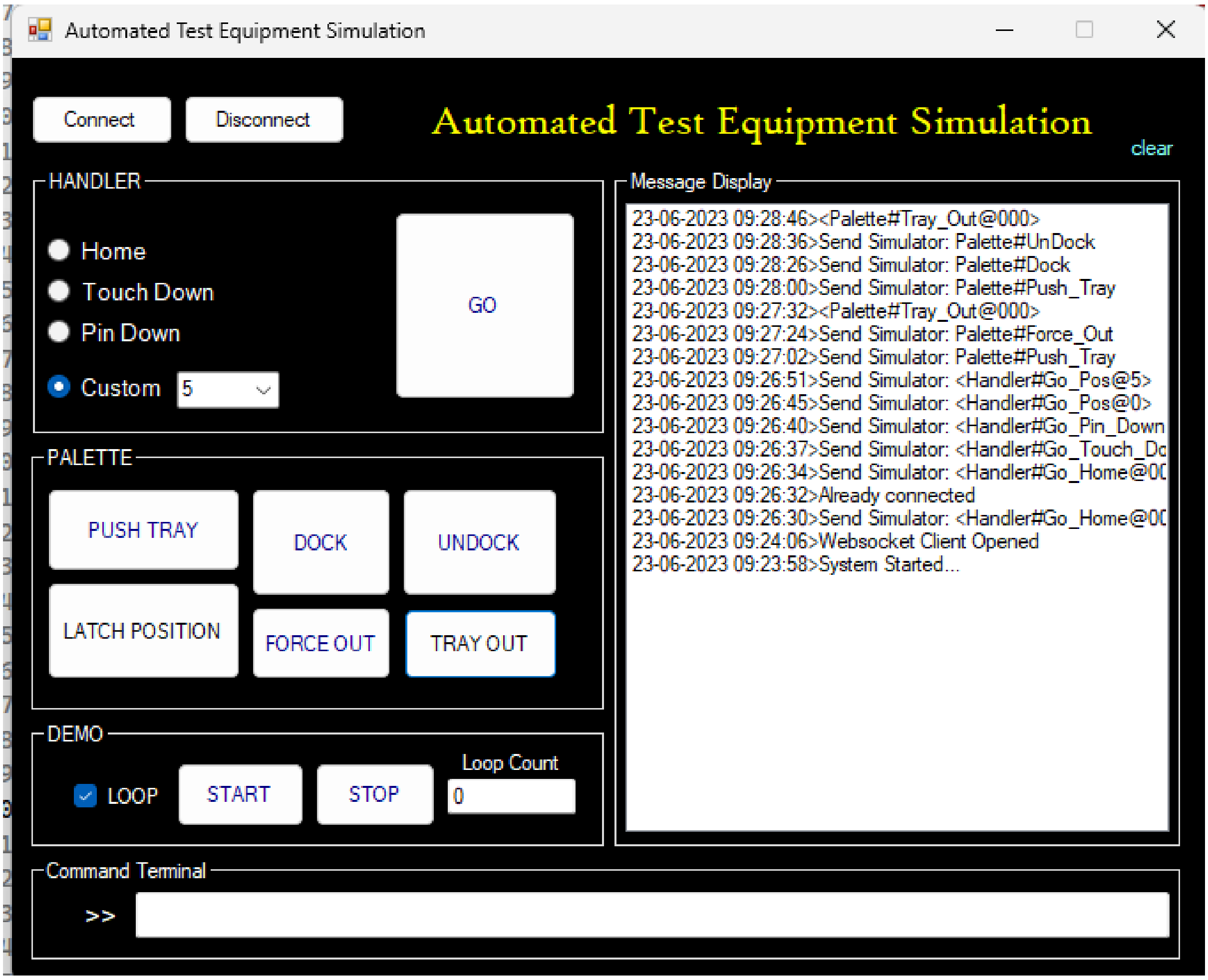

Fig. 3: Simulation Control Interface [Source: Author's]

Now let us discuss the control interface:

- **Connect**: The connect button connects with CoppeliaSim using Websocket port 8088. Within the same system, we use **ws://localhost:8088**. It will change when the simulator and controller are on different machines.
- **Disconnect**: This button is used to disconnect the simulator.
- **Handler group box**: a couple of user control available inside the group box. The "**G.O.**" button sends the command to the simulator. The various options are available for easy operation. The first option is "**Home**." To move the handler to the home position, select this radio button. The "**touch-down**" is where the handler touches the pogo pins. The "**Pin Down**" option presses a little distance and connects with DUT PCB. There is an option called custom. It is used for debugging or maintenance purposes. Select the position from the combo box and click the "**Go**" Button. The handler will reach the position where we want to.
- **Pallet group box**: there are four buttons inside it. The "**Push tray**" is used to push the tray toward the dock of the pallet. "Dock" means the system will dock the pallet if we press this button. The undock button is the opposite of the dock button. Once the DUT test is over, "undock" and bring the pallet outside. Pressing this button, start undocking. Sometimes pallet is stuck anywhere. To force out the pallet, we use the "**FORCEOUT**" button. When the power is cut, and the power restore.
- **DEMO group box:** the two buttons are available under this group box. Press the "**START**" button to start the demonstration automatically and the "**STOP**" button to stop at any time. The "Loop Count" text box shows the number of loops executed. If we want to run continuously, check on the "LOOP" check box.
- **Message display group box**: this part displays various messages. It helps complete system debugging.

- **Command terminal group box**: is used to send commands through typing. Instead of a button, it sends various commands to the simulator.

6. RECOMMENDATIONS :

- To be familiar with CoppeliaSim: https://www.srinivaspublication.com/journal/index.php/ijcsbe/article/view/1322/630. [11]
- The paper on Forward kinematics in CoppeliaSim: https://srinivaspublication.com/wp-content/uploads/2021/04/3.-Robot-Simulation_Fullpaper.pdf [12]
- The paper to build custom robotic arm in CoppeliaSim: https://srinivaspublication.com/wp-content/uploads/2021/04/4.-A-Custom-Robot-in_Fullpaper.pdf [13]
- The paper on Inverse kinematics in CoppeliaSim: https://srinivaspublication.com/wp-content/uploads/2021/05/7.-Inverse-5_Fullpaper.pdf [14]
- The paper to create a 3D printer in CoppeliaSim [15]: https://www.srinivaspublication.com/journal/index.php/ijaeml/article/view/1129/574
- This described model is not as actual as accurate. This a concept. According to our project requirement, we need to make fine-tuning.
- The simulation speed depends on the system configuration of the running system. A higher configuration is recommended for a smooth simulation.
- These works can be treated as example projects. Taking it, the researcher can customize the simulation or code part according to their need.

7. CONCLUSION :

We demonstrated a comprehensive framework for automated test equipment simulation in CoppeliaSim using C# over WebSocket. The framework offers a cost-effective, time-efficient solution for testing and validating electronic motherboards. It demonstrates high simulation accuracy, real-time communication capabilities, and seamless C# programming language integration. It offers cost savings, accelerates the time-to-market process, and provides flexibility in simulating various test equipment configurations and scenarios. It is recommended to further enhance the framework by improving simulation accuracy, expanding the test equipment library, supporting industry standards, and integrating with test automation tools. Validation against real-world data and collaboration within the research community will also contribute to the continuous improvement and adoption of the framework. It empowers engineers to achieve a cost-effective, efficient, and accurate testing solution.

REFERENCES :

[1] Miller, L. W., Stanton, R. E., & Crawford, G. B. (1984). DYNA-SIM: A Nonstationary Queuing Simulation with Application to the Automated Test Equipment Problem. RAND CORP SANTA MONICA CA. pp. 01-71. Google Scholar↗

[2] Montiel, C. M. (2012, November). Folded planar Marchand balun with improved isolation for radio frequency Automated Test Equipment applications. In 80th ARFTG Microwave Measurement Conference (pp. 1-3). IEEE. Google Scholar↗

[3] Barnes, H., Moreira, J., Ossining, H., Wollitzer, M., Schmid, T., & Tsai, M. (2006, December). Development of a pogo pin assembly and via design for multi-gigabit interfaces on automated test equipment. In 2006 Asia-Pacific Microwave Conference (pp. 381-384). IEEE. Google Scholar↗

[4] Boot, R., Richert, J., Schutte, H., & Rukgauer, A. (1999, August). Automated test of ECUs in a hardware-in-the-loop simulation environment. In Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No. 99TH8404) (pp. 587-594). IEEE. Google Scholar↗

[5] Jiang, C., Liu, B., Yin, Y., & Liu, C. (2009, July). Study real-time test scripts in automated test equipment. In 2009 8th International Conference on Reliability, Maintainability and Safety (pp. 738-742). IEEE. Google Scholar↗

[6] Moreira, J., Howieson, M., Petersen, M., Broman, M., & Kenton, J. (2006). Passive equalization of test fixtures for high-speed digital measurements with automated test equipment. In Proceedings of the 2006 International Design and Test Workshop, pp. 01-11. Google Scholar↗

[7] Liguori, F. (1974). Introduction to Current Computer Aids to Digital Test Design for Automated Test Equipment. Naval Air Engineering Center Lalehurst, NJ, pp. 01-20. Google Scholar↗

[8] Wang, Q., Chai, M., Wang, H., Zhang, H., Chai, J., & Lin, B. (2022, October). Cloud-Based Simulated Automated Testing Platform for Virtual Coupling System. In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) (pp. 2738-2743). IEEE. Google Scholar↗

[9] Gwóźdź, M. (2016, June). Reference current generator for high-power automated test equipment. In 2016 Progress in Applied Electrical Engineering (PAEE) (pp. 1-5). IEEE. Google Scholar↗

[10] Zaidi, Y., Fitz-Coy, N. G., & Zyl, R. V. (2019). Rapid, automated test, verification, and validation for the CubeSats. *International Journal of Space Science and Engineering, 5*(3), 242-268. Google Scholar↗

[11] Chakraborty,Sudip, & Aithal,P. S.,(2022).Conveyor Belt Speed Control Through CAN BUS in CoppeliaSim using Arduino Mega2560.International Journal of Case Studies in Business, IT, and Education (IJCSBE), 6(1), 194-201. Google Scholar↗

[12] Chakraborty, Sudip, & Aithal, P. S., (2021). Forward Kinematics Demonstration of 6DF Robot using CoppeliaSim and C#. *International Journal of Applied Engineering and Management Letters (IJAEML), 5*(1), 29-37. Google Scholar↗

[13] Chakraborty, Sudip, & Aithal, P. S., (2021). A Custom Robotic ARM in CoppeliaSim. *International Journal of Applied Engineering and Management Letters (IJAEML), 5*(1), 38- 50. Google Scholar↗

[14] Chakraborty, Sudip, & Aithal, P. S., (2021). An Inverse Kinematics Demonstration of a Custom Robot using C# and CoppeliaSim. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 5*(1), 78-87. Google Scholar↗

[15] Sudip Chakraborty& Aithal, P. S., (2022). A Simulated 3D Printer in CoppeliaSim. *International Journal of Applied Engineering and Management Letters (IJAEML), 6*(1), 22-32. Google Scholar↗

****